WHAT IS CLAIMED IS:

1. A memory controller for a memory device, the memory controller comprising a lossless decompression accelerator operative to decompress data that has been externally compressed and loaded into the memory device.

2. The memory controller as recited in claim 1, wherein the memory controller is for a flash memory device.

3. A memory device comprising the memory controller as recited in claim 1.

4. A memory device comprising a lossless decompression accelerator operative to decompress data that has been externally compressed and loaded into the memory device.

5. The memory device as recited in claim 4, wherein the memory device is a flash memory device.

6. A decompression accelerator for decompressing Lempel-Ziv-Huffman compressed data from an input stream and sending decompressed data corresponding thereto to an output stream, the decompression accelerator comprising:

(a) a variable-length token decoder for selectively decoding Huffman-encoded code portions of Lempel-Ziv tokens and for selectively retrieving and passing extra bit portions of said Lempel-Ziv tokens without Huffman decoding; and

(b) a Lempel-Ziv decoder for decompressing Lempel-Ziv tokens obtained from said variable-length token decoder.

7. The decompression accelerator as recited in claim 6, wherein said variable-length token decoder includes:

i) a bit buffer, for breaking fixed-length words from the input bit stream into variable-length words for Huffman decoding and for retrieving extra bits, said bit buffer having a variable-length output;

ii) a token analyzer, for determining characteristics of a Lempel-Ziv token, for coordinating said selective Huffman decoding of said code portion of each said Lempel-Ziv token, and, if said each Lempel-Ziv token includes said extra bit portion, for coordinating said selective passing of said extra bit portion without Huffman decoding; and

iii) .a Huffman decoder for effecting said selective Huffman decoding.

8. A decompression accelerator as recited in claim 7, wherein said Huffman decoder is operative to perform dynamic Huffman decoding.

9. A decompression accelerator as recited in claim 7, further comprising at least one component selected from the group consisting of:

(c) an input buffer, for buffering input data words between the input stream and said bit buffer;

(d) a bit buffer controller, for controlling operation of said bit buffer;

(e) an output selector, for selecting output of said Huffman decoder and for selectively passing said variable-length output of said bit buffer; and

(f) a token constructor, for, for each said Lempel-Ziv token that includes said extra bit portion, assembling said code portion of said each

Lempel-Ziv token as decoded by said Huffman decoder and said extra bit portion so as to reproduce said each Lempel-Ziv token.

10. A decompression accelerator as recited in claim 6, further comprising at least one component selected from the group consisting of:

(c) a token buffer for buffering said Lempel-Ziv tokens from said variable-length token decoder and from said Lempel-Ziv decoder; and

(d) an output buffer, for buffering decompressed output from said Lempel-Ziv decoder to the output stream.

11. A data processor comprising a decompression accelerator as recited in claim 6.

12. A data processor as recited in claim 11, further comprising flash memory.

13. A memory controller comprising a decompression accelerator as recited in claim 6.

14. The memory controller recited in claim 13, wherein the memory controller is a flash memory controller.

15. A memory device comprising the memory controller recited in claim 13.

16. A memory device comprising a decompression accelerator as recited in claim 6.

17. The memory device as recited in claim 16, wherein the memory device is a flash memory device.

18. A method for efficiently storing and retrieving data for use, comprising:

(a) providing a data processor having data storage and having a decompression accelerator;

(b) compressing the data according to a lossless data compression format, thereby producing compressed data;

(c) writing said compressed data to said processor data storage; and

(d) decompressing said compressed data stored in said processor data storage using said processor decompression accelerator.

19. The method as recited in claim 18, wherein said compressing is effected using lossless data compression software.

20. The method as recited in claim 18, wherein said lossless data compression format is selected from the group consisting of:

i) Lempel-Ziv format;

ii) Lempel-Ziv-Static Huffman format; and

iii) Lempel-Ziv-Dynamic Huffman format.

21. The method as recited in claim 19, wherein said lossless data compression software is selected from the group consisting of:

iv) Lempel-Ziv compression software;

v) Lempel-Ziv-Static Huffman compression software; and

vi) Lempel-Ziv-Dynamic Huffman compression software.